

TITLE

**APPARATUS FOR CONTROLLING PRINTER TO IMPROVE PRINTING
SPEED AND METHOD THEREOF**

CLAIM OF PRIORITY

[0001] This application makes reference to, incorporates the same herein, and claims all benefits accruing under 35 U.S.C. §119 from my application entitled *APPARATUS FOR CONTROLLING PRINTER TO IMPROVE PRINTING SPEED AND METHOD THEREOF* filed with the Korean Industrial Property Office on 10 January 2001 and there duly assigned Serial No. 1346/2001.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention relates to a printer control system, and more particularly, to an apparatus for controlling a printer, by which an image is printed in a raster mode by counting functions, which affect the printing speed, from an image, and a method thereof.

Description of the Related Art

[0003] Generally, printer drivers include a font processing module and an image processing module, and control printing operations. Here, an image processing module includes a vector module and a raster (or bitmap) module. A vector module defines each image written in a

document as objects and produces commands corresponding to the objects. A raster module converts an image into a bitmap in contrast with in a vector module and transmits the bitmap to a printer.

[0004] In a process for processing image data in a conventional printer driver, when an application executes a printing command to print a document, the graphic device interface (GDI) of the Windows calls the printer driver. Next, the printer driver sets a vector mode, in which a document can be more precisely printed, rather than a raster mode as a default, when typical documents are printed, and processes an image on an object-by-object basis. Here, an image included in the document of a specific application is made up of hundreds of thousands of objects.

[0005] Accordingly, the printer driver must produce a large number of vector commands if it converts these objects into vector commands. Also, when a document having complicate data is printed, a printer must interpret a large number of objects received from the printer driver and reorganize them into one page, resulting in a time-consuming printing for one page.

SUMMARY OF THE INVENTION

[0006] To solve the above problems, an objective of the present invention is to provide a method of controlling a printer, by which the printing speed is improved by counting the number of functions for outputting objects in image data and converting a page area corresponding to the counted functions into a bitmap if the number of functions is equal to or greater than a

predetermined limit value.

[0007] Another object of the present invention is to provide an apparatus for controlling a printer using the printer controlling method.

[0008] It is further an object of the present invention to produce an apparatus and a method for producing font commands, raster commands, and vector commands for a printer so as to maximize printing speed.

[0009] It is also an object of the present invention to produce an apparatus and method wherein raster commands or bitmaps are submitted to the printer whenever the total number of vector object functions exceeds a predetermined number.

[0010] To achieve the first, third and fourth objectives, the present invention provides a printer controlling method including (a) classifying a page area to be printed into image data and font data, when a printing command is received from an application; (b) dividing the image data according to objects and processing the image data in units of objects; and (c) counting the number of functions for outputting the objects processed in step (b), and converting the page area into a bitmap if the counted value is equal to or greater than a limit value.

[0011] To achieve the second, third and fourth objectives, the present invention provides a printer controlling apparatus including a graphic engine unit for dividing a printing page received from an application into image data and font data; a printer driver for dividing the image data into objects and producing a bitmap command if the number of functions depending on objects is equal to or greater than a limit value or producing an object-unit command if the number of

functions is smaller than the limit value; and a printer for printing the page according to the command produced by the printer driver.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] A more complete appreciation of the invention, and many of the attendant advantages thereof, will be readily apparent as the same becomes better understood by reference to the following detailed description when considered in conjunction with the accompanying drawings in which like reference symbols indicate the same or similar components, wherein:

[0013] FIG. 1 is a block diagram of the entire structure of a printer controlling apparatus according to the present invention;

[0014] FIGS. 2A and 2B are detailed block diagrams of the print driver of FIG. 1; and

[0015] FIG. 3 is a flowchart illustrating a printer controlling method according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0016] Referring to FIG. 1, a printer controlling apparatus according to the present invention is made up of an application unit 110, a graphic engine unit 120, a printer driver 130 and a printer 140. The application unit 110, which is a program developed to perform a specific operation using a computer, generates a data processing command such as a font, a graphic or an image depending on the characteristics of an operation for which the program has been designed. The

1 graphic engine unit 120 includes a graphic device interface (GDI) and a device driver interface
2 (DDI), and divides data from the application unit 110 into an image and a font and generates
3 information on the image and information on the font.

4 **[0017]** The printer driver 130 receives image data and font data divided by the graphic engine
5 unit 120, processes each of the data and converts the data into a command capable of being
6 recognized by a printer, for example, into a printer control language or a postscript. Here, the
7 printer driver 130 processes the image data in a vector mode and in a raster mode. In a vector
8 mode, image data is divided into objects, and each of the objects is produced into a command.
9 For example, if a document has an image of a circle and a rectangle, the printer driver 130
10 divides the image into objects of a circle and a rectangle, defines the coordinate, property and
11 color of each of the objects, and converts each of the objects into a command corresponding to
12 the defined image information items of the object. Also, in a raster mode, the image data is
13 converted into a bitmap.

14 **[0018]** Also, the printer driver 130 individually counts functions for outputting the objects
15 from a document, and processes the image data as a bitmap in a raster mode if the counted value
16 is equal to or greater than a limit value, or processes the image data in a vector mode if the
17 counted value is smaller than the limit value. The printer 140 interprets commands associated
18 with the image and font received from the printer driver 130 and produces printable data having a
19 resolution, for example, 600dpi or 1200dpi.

20 **[0019]** FIGS. 2A and 2B are detailed block diagrams of the print driver of FIG. 1. Referring

1 to FIGS. 2A and 2B, the printer driver 130 is made up of a graphic module unit 212 for
2 processing image data and a font module unit 222 for processing font data. The graphic module
3 unit 212 is made up of a vector module unit 213 and a raster module unit 214. To be more
4 specific, the vector module unit 213 includes a path module unit 215 for dividing the image data
5 into objects and producing commands for the objects, and a brush module unit 216 for dividing
6 the image data by colors and producing commands for the colors. The raster module unit 214
7 includes a bit map module unit 217 for converting image data into a bitmap and producing a
8 command for the bitmap. The font module unit 222 includes a first font type unit 223 which
9 corresponds to a download truetype font, a second font type unit 224 which corresponds to a
10 download bitmap font, and a third font type unit 225 which corresponds to a truetype font which
11 is graphic.

12 **[0020]** FIG. 3 is a flowchart illustrating a printer controlling method according to the present
13 invention. First, an application generates a printing command at a user's request, in step 310.
14 Next, the graphic engine unit 120 divides the area of a printing page into image data and font
15 data when the printing command is received from the application, in step 320. Then, the printer
16 driver 130 performs the following printing command by determining whether the page area is the
17 image data or the font data, in step 330.

18 **[0021]** That is, if it is determined in step 330 that the page area is image data, it is determined
19 whether the graphic mode selected by a user is a vector mode, in step 340. If it is determined
20 that the document area is not image data, the document area is determined to be the font data.

Thus, a font command is generated, in step 335, and then it is transferred to a printer, in step 390.

Then, if it is determined that the graphic mode is a vector mode, a vector mode in which image data is divided according to objects and processed as an object-unit command is executed, in step 350. Otherwise, a raster command is produced by generating a bitmap, in step 375.

[0022] After the step 350, if it is determined in step 360 that there are vector object functions for outputting each of the objects, the number of functions is counted, in step 370, and then the counted value is compared with a predetermined limit value, in step 372. If it is determined in step 360 that there are no vector object functions for outputting each of the objects, only areas having no objects are processed as a bitmap using a raster mode, in step 362. In a raster mode, an image is converted into a bitmap without dividing the image into objects, in contrast with a vector mode. Here, an example of an object function written in C Language to output an object is provided as follows:

```

    BOOL DrvStrokePath (
        IN SURFOBJ *pso,
        IN PATHOBJ *ppo,
        IN CLIPOBJ *pco,
        IN XFORMOBJ *pxo,
        IN BRUSHOBJ *pbo,
        IN POINTL *pptlBrushOrg,
        IN LINEATTRS *plineattrs,
    )

```

IN MIX mix

);

[0023] DrvStrokePath strokes a path when called by GDI. If the driver has hooked the function, and if the appropriate GCAP's are set, GDI calls DrvStrokePath when GDI draws a line or curve with any set of attributes. Parameter "pso" identifies the surface on which to draw. Parameter "ppo" points to a PATHOBJ structure. GDI PATHOBJ_Xxx service routines are provided to enumerate the lines, Bezier curves, and other data that make up the path. This indicates what is to be drawn. Parameter "pco" points to a CLIPOBJ structure. GDI CLIPOBJ_Xxx service routines are provided to enumerate the clip region as a set of rectangles. Optionally, all the lines in the path may be enumerated preclipped by CLIPOBJ. This means that drivers can have all their line clipping calculations done for them. Parameter "pxo" points to a XFORMOBJ. This is only needed when a geometric wide line is to be drawn. It specifies the transform that maps world coordinates to device coordinates. This is needed because the path is provided in device coordinates but a geometric wide line is actually widened in world coordinates. The XFORMOBJ can be queried to find the transform. Parameter "pbo" specifies the brush to be used when drawing the path. Parameter "pptlBrushOrg" points to the brush origin used to align the brush pattern on the device. Parameter "plineattrs" points to a LINEATTRS structure. Note that the elStyleState member must be updated as part of this function if the line is styled. Also not that the ptlLastPel member must be updated if a single pixel with cosmetic line is being drawn. Parameter "mix" specifies how to combine the brush

1 with the destination.

2 **[0024]** Thereafter, if the counted value is equal to or greater than the predetermined limit
3 value, a raster command for processing a page area corresponding to the counted functions as a
4 bitmap is produced, in step 375. Otherwise, a vector command for processing the page area in
5 units of objects is produced, in step 380. Thus, when bitmap data is produced using a raster
6 mode, data is significantly simplified, and the printing speed is also significantly increased.
7 Then, the vector command, the raster command and the font command are transferred to a
8 printer, in step 390.

9 **[0025]** Although the invention has been described with reference to a particular embodiment,
10 it will be apparent to one of ordinary skill in the art that modifications of the described
11 embodiment may be made without departing from the spirit and scope of the invention.

12 **[0026]** According to the present invention as described above, when a document having
13 complicate data is printed, a printer driver automatically converts a vector mode set as a default
14 into a raster mode, whereby the time to analyze data and form an image and the overflow of
15 memory, which may be generated due to the above operations, are reduced. Therefore, the
16 printing speed of a printer is improved, and the size of memory can be reduced.

17 **[0027]** It should be understood that the present invention is not limited to the particular
18 embodiment disclosed herein as the best mode contemplated for carrying out the present
19 invention, but rather that the present invention is not limited to the specific embodiments
20 described in this specification except as defined in the appended claims.